

讲堂 > Linux性能优化实战 > 文章详情

01 | 如何学习Linux性能优化?

2018-11-21 倪朋飞



01 | 如何学习Linux性能优化?

朗读人：冯永吉 10'34" | 4.84M

你好，我是倪朋飞。

你是否也曾跟我一样，看了很多书、学了很多 Linux 性能工具，但在面对 Linux 性能问题时，还是束手无策？实际上，性能分析和优化始终是大多数软件工程师的一个痛点。但是，面对难题，我们真的就无解了吗？

固然，性能问题的复杂性增加了学习难度，但这并不能成为我们进阶路上的“拦路虎”。在我看来，大多数人对性能问题“投降”，原因可能只有两个。

一个是你没找到有效的方法学原理，一听到“系统”、“底层”这些词就发怵，觉得东西太难，自己一定学不会，自然也就无法深入学下去，从而不能建立起性能的全局观。

再一个就是，你看到性能问题的根源太复杂，既不懂怎么去分析，也不能抽丝剥茧找到瓶颈。

你可能会想，反正程序出了问题，上网查就是了，用别人的方法，囫圇吞枣地多试几次，有可能就解决了。于是，你懒得深究这些方法为啥有效，更不知道为什么，很多方法在别人的环境有

效，到你这儿就不行了。

所以，相同的错误重复在犯，相同的状况也是重复出现。

其实，性能问题并没有你想像得那么难，**只要你理解了应用程序和系统的少数几个基本原理，再进行大量的实战练习，建立起整体性能的全局观**，大多数性能问题的优化就会水到渠成。

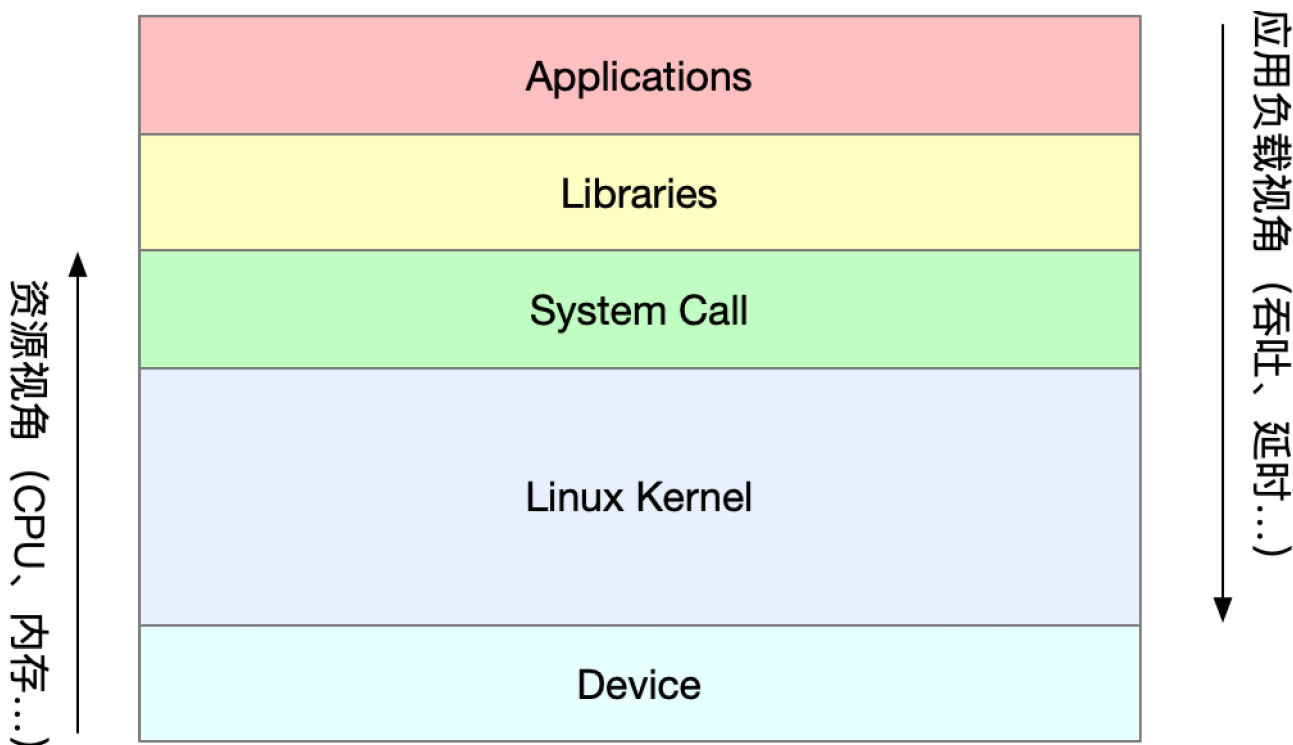
我见过很多工程师，在分析应用程序所使用的第三方组件的性能时，并不熟悉这些组件所用的编程语言，却依然可以分析出线上问题的根源，并能通过一些方法进行优化，比如修改应用程序对它们的调用逻辑，或者调整组件的配置选项等。

还是那句话，**你不需要了解每个组件的所有实现细节**，只要能理解它们最基本的工作原理和协作方式，你也可以做到。

性能指标是什么？

学习性能优化的第一步，一定是了解“性能指标”这个概念。

当看到性能指标时，你会首先想到什么呢？我相信“**高并发**”和“**响应快**”一定是最先出现在你脑海里的两个词，而它们也正对应着性能优化的两个核心指标——“**吞吐**”和“**延时**”。这两个指标是**从应用负载的视角**来考察性能，直接影响了产品终端的用户体验。跟它们对应的，是**从系统资源的视角**出发的指标，比如资源使用率、饱和度等。



我们知道，随着应用负载的增加，系统资源的使用也会升高，甚至达到极限。而**性能问题的本质**，就是系统资源已经达到瓶颈，但请求的处理还不够快，无法支撑更多的请求。

性能分析，其实就是**找出应用或系统的瓶颈，并设法去避免或者缓解它们**，从而更高效地利用系统资源处理更多的请求。这包含了一系列的步骤，比如下面这六个步骤。

- 选择指标评估应用程序和系统的性能；
- 为应用程序和系统设置性能目标；
- 进行性能基准测试；
- 性能分析定位瓶颈；
- 优化系统和应用程序；
- 性能监控和告警。

了解了这些性能相关的基本指标和核心步骤后，该怎么学呢？接下来，我来说说要学好 Linux 性能优化的几个重要问题。

学这个专栏需要什么基础

首先你要明白，我们这个专栏的核心是**性能的分析 and 优化**，而不是最基本的 Linux 操作系统的使用方法。

因而，我希望你最好用过 Ubuntu 或其他 Linux 操作系统，然后要具备一些**编程基础**，比如：

- 了解 Linux 常用命令的使用方法；
- 知道怎么安装和管理软件包；
- 知道怎么通过编程语言开发应用程序等。

这样，在我讲性能时，你就更容易理解性能背后的原理，特别是在结合专栏里的案例实践后，对性能分析能有更直观的体会。

这个专栏不会像教科书那样，详细教你操作系统、算法原理、网络协议乃至各种编程语言的全部细节，但一些重要的系统原理还是必不可少的。我还会用实际案例一步步教你，贯穿从应用程序到操作系统的各个组件。

学习的重点是什么？

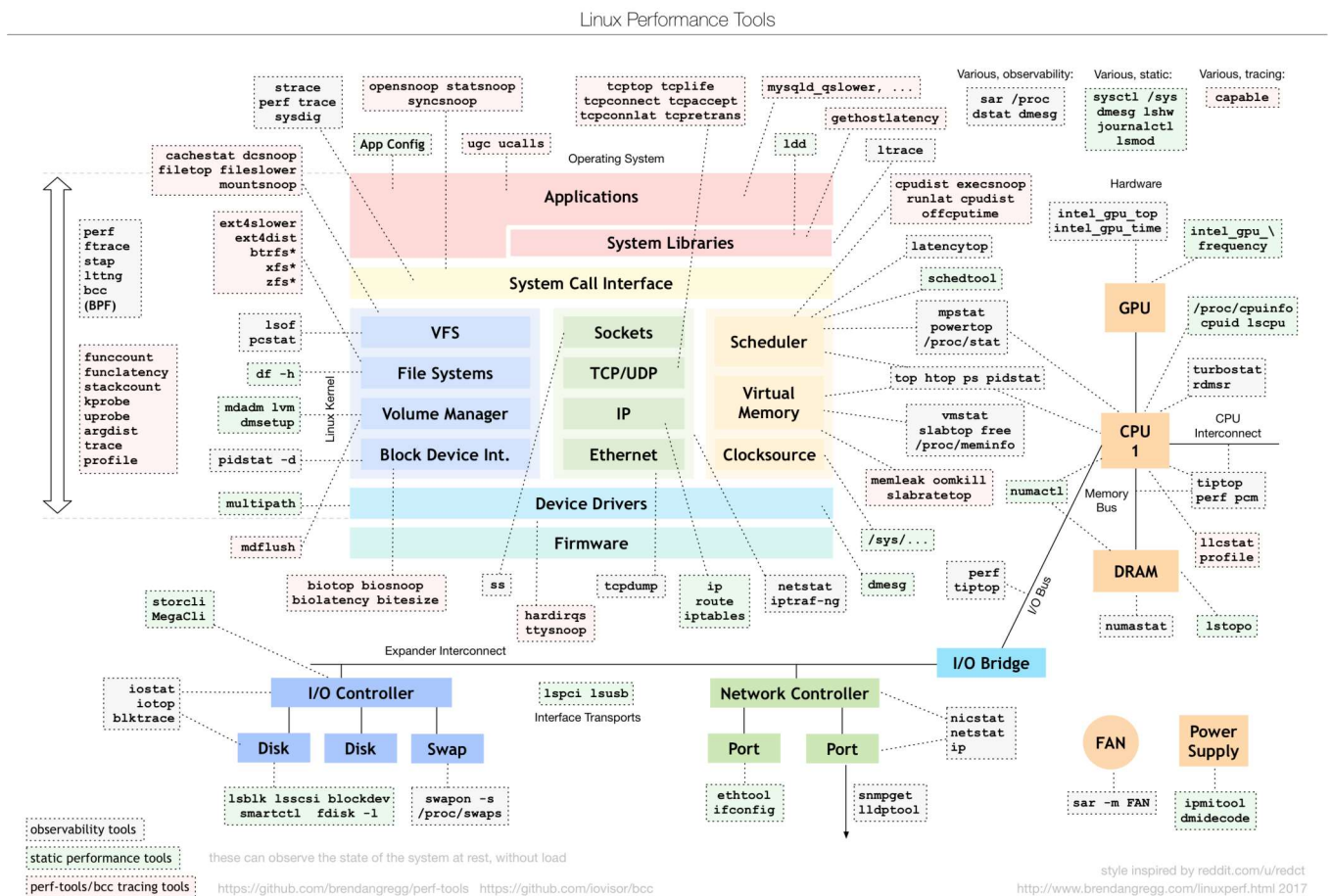
想要学习好性能分析和优化，**建立整体系统性能的全局观**是最核心的话题。因而，

- 理解最基本的几个系统知识原理；
- 掌握必要的性能工具；

- 通过实际的场景演练，贯穿不同的组件。

这三点，就是我们学习的重中之重。我会在专栏的每篇文章中，针对不同场景，把这三个方面给你讲清楚，你也一定要花时间和心思来消化它们。

其实说到性能工具，就不得不提性能领域的大师布伦丹·格雷格（Brendan Gregg）。他不仅是动态追踪工具 DTrace 的作者，还开发了许许多多的性能工具。我相信你一定见过他所描绘的 Linux 性能工具图谱：



(图片来自**brendangregg.com**)

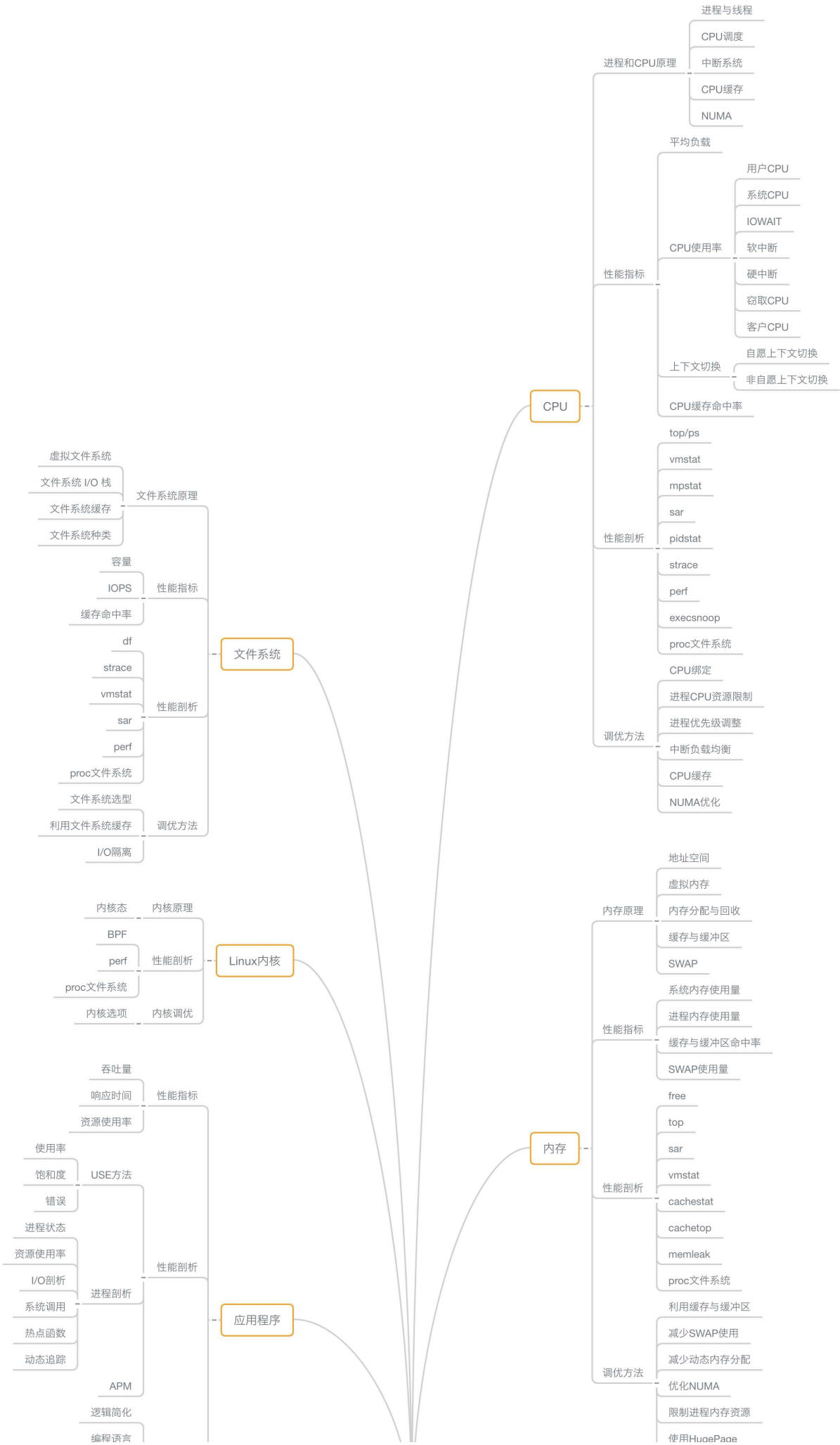
这个图是 Linux 性能分析最重要的参考资料之一，它告诉你，在 Linux 不同子系统出现性能问题后，应该用什么样的工具来观测和分析。

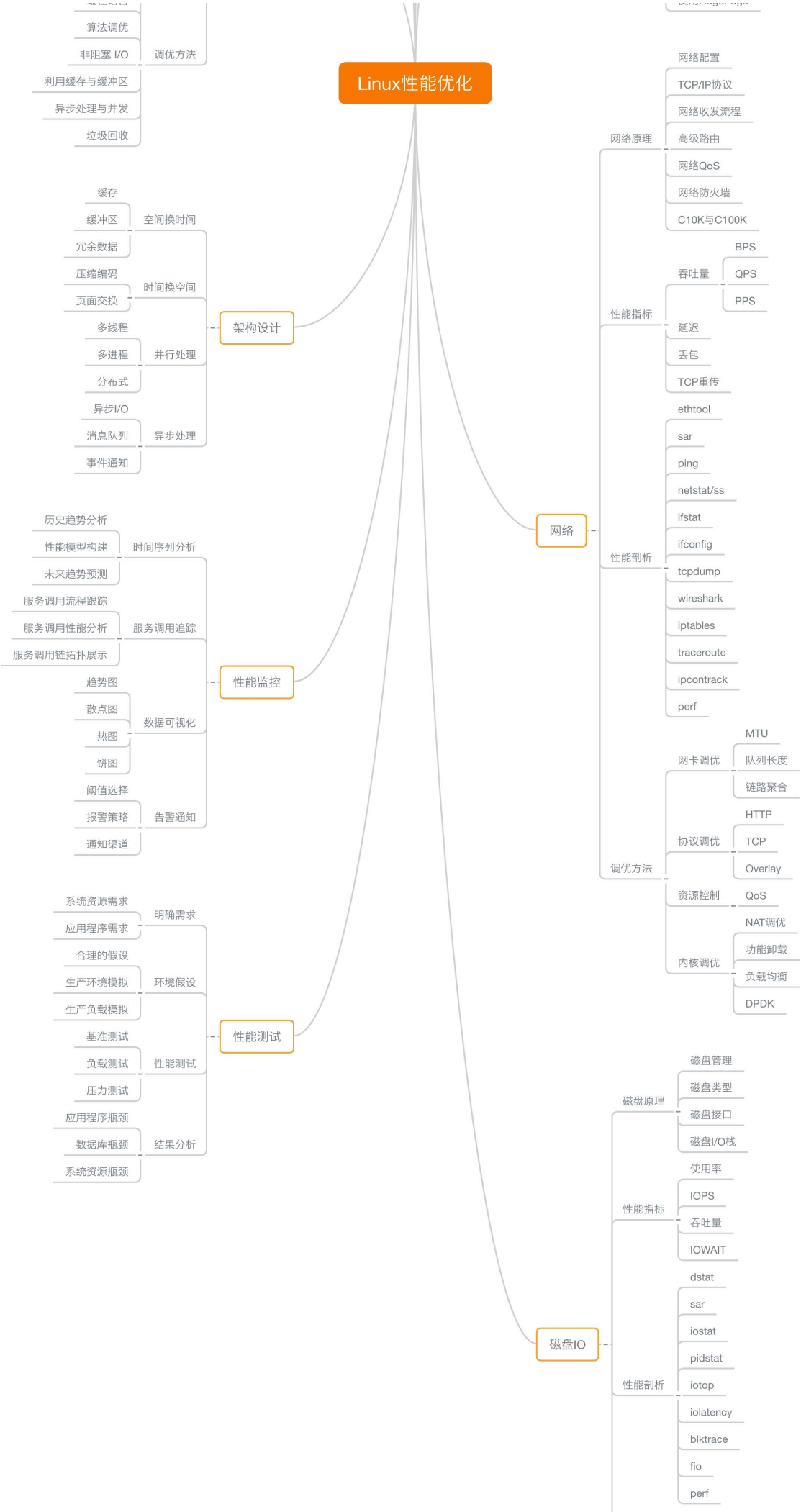
比如，当遇到 I/O 性能问题时，可以参考图片最下方的 I/O 子系统，使用 iostat、iotop、blktrace 等工具分析磁盘 I/O 的瓶颈。你可以把这个图保存下来，在需要的时候参考查询。

另外，我还要特别强调一点，就是**性能工具的选用**。有句话是这么说的，一个正确的选择胜过千百次的努力。虽然夸张了些，但是选用合适的性能工具，确实可以大大简化整个性能优化过程。在什么场景选用什么样的工具、以及怎么学会选择合适工具，都是我想教给你的东西。

但是切记，**千万不要把性能工具当成学习的全部**。工具只是解决问题的手段，关键在于你的用法。只有真正理解了它们背后的原理，并且结合具体场景，融会贯通系统的不同组件，你才能真正掌握它们。

最后，为了让你对性能有个全面的认识，我画了一张思维导图，里面涵盖了大部分性能分析和优化都会包含的知识，专栏中也基本都会讲到。你可以**保存或者打印**下来，每学会一部分就标记出来，记录并把握自己的学习进度。







怎么学更高效？

前面我给你讲了 Linux 性能优化的学习重点，接下来我再跟你分享一下，我的几个学习技巧。掌握这些技巧，可以让你学得更轻松。

技巧一：虽然系统的原理很重要，但在刚开始一定不要试图抓住所有的实现细节。

深陷到系统实现的内部，可能会让你丢掉学习的重点，而且繁杂的实现逻辑，很可能会打退你学习的积极性。所以，我个人观点是一定要适度。

你可以先学会我给你讲的这些系统工作原理，但不要去深究 Linux 内核是如何做到的，而是要把你的重点放到如何观察和运用这些原理上，比如：

- 有哪些指标可以衡量性能？
- 使用什么样的性能工具来观察指标？
- 导致这些指标变化的因素等。

技巧二：边学边实践，通过大量的案例演习掌握 Linux 性能的分析 and 优化。

只有通过机器上练习，把我讲的知识 and 案例自己过一遍，这些东西才能转化成你的。我精心设计这些案例，正是为了让你有更好的学习理解 and 操作体验。

所以我强烈推荐你去实际运行、分析这些案例，或者用学到的知识去分析你自己的系统，这样你会有更直观的感受，获得更好的学习效果。

技巧三：勤思考，多反思，善总结，多问为什么。

想真正学懂一门知识，最好的方法就是问问题。当你能提出好的问题时，就说明你已经深入了解了它。

你可以随时在留言区给我留言，写下自己的疑问、思考和总结，和我还有其他的学习者一起讨论切磋。你也可以写下自己经历过的性能问题，记录你的分析步骤 and 优化思路，我们一起互动探讨。

学习之前，你的准备

作为一个包含大量案例实践的课程，我会在每篇文章中，使用一到两台 Ubuntu 18.04 虚拟机，作为案例运行和分析的环境。如果你只是单纯听音频的讲解，却从不动手实践，学习的效果一定会大打折扣。

所以，你是不是可以准备好一台 Linux 机器，用于课程案例的实践呢？**任意的虚拟机或物理机都可以，并不局限于 Ubuntu 系统。**

思考

今天的内容是我们后续学习的热身准备。从下篇文章开始，我们就要正式进入 Linux 性能分析和优化了。所以，我想请你来聊一聊，你之前在解决 Linux 性能问题时，有遇到过什么样的困难或者疑惑吗？或者是之前自己学习 Linux 性能优化时，有哪些问题吗？参考我今天所讲的内容，你又打算怎么来学这个专栏？

欢迎在留言区和我分享。



©版权归极客邦科技所有，未经许可不得转载

上一篇 开篇词 | 别再让Linux性能问题成为你的绊脚石

下一篇 02 | 基础篇：到底应该怎么理解“平均负载”？

写留言

精选留言



forever

63

我遇到性能瓶颈的排查思路

有监控的情况下，首先去看看监控大盘，看看有没有异常报警，如果初期还没有监控的情况我会按照下面步骤去看看系统层面有没有异常

- 1、我首先会去看看系统的平均负载，使用top或者htop命令查看,平均负载体现的是系统的一个整体情况，他应该是cpu、内存、磁盘性能的一个综合，一般是平均负载的值大于机器cpu的核数，这时候说明机器资源已经紧张了
- 2、平均负载高了以后，接下来就要看看具体是什么资源导致，我首先会在top中看cpu每个核的使用情况，如果占比很高，那瓶颈应该是cpu,接下来就要看看是什么进程导致的
- 3、如果cpu没有问题，那接下来我会去看内存，首先是用free去查看内存的是用情况，但不直接看他剩余了多少，还要结合看看cache和buffer，然后再看看具体是什么进程占用了过高的内存，我也是用top去排序
- 4、内存没有问题的话就要去看磁盘了，磁盘我用iostat去查看，我遇到的磁盘问题比较少
- 5、还有就是带宽问题，一般会用iftop去查看流量情况，看看流量是否超过的机器给定的带宽
- 6、涉及到具体应用的话，就要根据具体应用的设定参数来查看，比如连接数是否查过设定值等
- 7、如果系统层各个指标查下来都没有发现异常，那么就要考虑外部系统了，比如数据库、缓存、存储等

基本上就上面这些步骤，有些不完整，希望跟着老师学习一些更系统的排查思路！

2018-11-21

作者回复

总结的很好👍

2018-11-21



C家族铁粉儿

D2打卡

👍 21

1. 笔记

技巧一：虽然系统的原理很重要，但在刚开始一定不要试图抓住所有的实现细节。”

深陷到系统实现的内部，可能会让你丢掉学习的重点，而且繁杂的实现逻辑，很可能会打退你学习的积极性。所以，我个人观点是一定要适度。

2. 心得

作为一个完美主义者，一学起原理类的东西，真的不要太容易跑偏😓经常是看着某个重要原理，就想着找找看相关内容，然后就各种跳转搜索，以前最开始学数据结构的定义，都能跑到编译原理上，最后开始计算二进制了。

有时候大半天了，一个原理都没看完，就各种死抠和联想。这么做确实印象深刻，但是真的很低效，心累。

老师这里说的适度，真的很重要，而且这个度，确实应该是过来人才知道啊。

我一向喜欢系统化的学习，能有个“升级简化版”的系统知识图谱，不要太开心。可惜不能上传图片，不然可以把每次标记和补充也都打个卡了。

开始学了，加油！冲着我的四个月涨工资的目标去了！

2018-11-21

作者回复

总结的好👍

2018-11-21



发条橙子。

👍 18

以前看服务器的资源使用只会简单的使用 top命令 看cpu使用的百分比。，但是却不清楚到底多高才算高危，面对持续增长我该怎么预防或处理，load指标具体的含义和cpu有什么关联..这些都没有一个整体的概念

2018-11-21

作者回复

下一篇就是平均负载😁

2018-11-21



Allen

👍 8

『day1』

这周工作中遇到了一个紧急的问题（用的是arm系列的单板），单板的空间几乎快满了。使用了top和free命令查看，单板内存的使用情况，仅仅凭借这两个命令，是不可能分析出来原因的。

查看/proc/<pid>/下的的meminfo、status等文件可以具体才看到虚拟内存和实际物理内存的使用情况。之前根本不了解/proc/<pid>里面的文件都是干嘛的。

希望跟着老师的专栏，可以了解下linux系统的基本知识，以后遇到相关问题时，可以有一些思路。

2018-11-21

作者回复

内存模块会详细介绍

2018-11-21



Geek_60821e

👍 7

希望之后课程能深入的用工具分析原因，不要讲工具怎么用，怎么用工具上网free的实例很多，找出瓶颈的原因是来听课目的，wish nice

2018-11-21



石壹笑

👍 7

切记，千万不要把性能工具当成学习的全部。
应消除干扰，专注于愿问题本身。

2018-11-21



mj4ever

👍 5

之所以选择学习这个专栏，就是希望能解决实际工作中的一些技术问题，当公司研发的产品在现场运行，出现性能问题的时，不会束手无策，毫无思路，误打误撞的去解决问题。

因此，希望通过3个月的学习，自己可以掌握以下几个方面：

- 1、建立整体性能的全局观
- 2、理解最基本的几个系统知识原理
- 3、掌握Linux 性能工具图谱的熟练应用

感谢大家，也感谢能不断坚持的自己。

2018-11-21

| 作者回复



2018-11-21



Dragon

👍 4

能否在翻页的左侧或右侧加一个暂停键, 你想如果边听边看文字时候暂停还要返回头顶,多麻烦

2018-11-23



Gopher

👍 3

小哥哥声音好听，语音很清晰

2018-11-22



上善若水

👍 3

之前调试网络性能问题，组网比较复杂，测试仪千万级用户拨号，越到后面越是上线速率急速下降，每个点都进行抓包监控丢包率，top也观测进程cpu利用率，就是找不出原因，进程cpu利用率也不高，丢包也不严重，对比抓包，发现上送包处理有很大延迟，后来pstack应用进程发现一处一点，原来应用进程有一处调用函数是阻塞的，光有生产者没有消费者，导致大用户拨号后阻塞超时，而且那个api还是第三方的没有源码，找到维护的组才确认问题，前后加班熬了两个星期才确认问题，所以痛下心来学习下性能定位课程

2018-11-21



夜空中最亮的星（华仔）

👍 3

已经在朋友圈转发2次老师的课了。

2018-11-21



梁中华

👍 3

打卡，希望作者必要的时候能结合虚拟机和docker来讲下，在这两种环境下的性能问题分析有什么要注意的，毕竟现在的应用不是运行在虚拟机就是运行在容器中

2018-11-21

| 作者回复

会的，案例也会用到docker

2018-11-21



代码GG陆晓明

👍 3

学习这块，是想应用到手机系统上。安卓的性能一直是个诟病，想通过一些工具进行复测，能够找到一些思路进行优化，然后使用一套测试打分，来评估到底是改善还是没有。

在了解学习linux的常用性能分析工具，想从本课程琢磨出一套方式，来进行系统的优化，找到一条优化安卓系统的有效方法。

感谢！

2018-11-21



燕羽阳

👍 2

性能问题考验的是综合能力和全局观，希望能和大家共同进步

2018-11-21



李建贵

👍 2

之前对Linux的仅停留在命令行的使用，通过这个专栏从性能视角深入的学习和了解Linux，相信学了这个对我的成长很有帮助！

2018-11-21



番茄炒鸡蛋

👍 2

前段时间遇到服务请求过多，导致内存增高，但是过段时间请求下降，但内存还是一直比较高，如果后续再次遇到请求过多，就会导致没有内存，程序崩溃，请问这类问题如何排查？

2018-11-21



潜。

👍 2

性能指标概念：高并发 => 吞吐 响应快 => 延时

该概念是从应用负载的角度出发：Application ▸ Libraries ▸ System Call ▸ Linux Kernel ▸ Drive

与之对应的是系统资源视角出发：Drive ▸ Linux Kernel ▸ System Call ▸ Libraries ▸ Application

性能指标的评判有以上二种常用的角度

接着六步

1.选择性能指标评估应用和系统的性能

2.为应用和系统设定性能目标

3.进行性能基准测试

4.性能分析定位瓶颈

5.优化系统和应用程序

6.性能监控和告警

六步总结，从正确的角度出发，设定目标（性能优化不是漫无目的的），基准测试（了解现有系统应用的运行时情况），根据情况分析瓶颈，优化它，设置监控和告警（其实可以再扩展比如达到一定的负载，采取降级等操作）

2018-11-21

作者回复

总结的很好

2018-11-21



King_stone

👍 2

遇到过的问题：Linux上tomcat性能下降，资源消耗较大，最终判断是JAVA程序一个方法的问题。解决和监控的手段不够智能，写过一个监控但是JAVA本身的监控不稳定，消耗也较大，最终撤掉，希望课程能对这方面优化监控有介绍

2018-11-21



java小白

👍 2

D2打卡，做了一段时间的性能优化工作，发现自己工作中只是重复的进行压测，压接口压场景，遇到的最大问题是CPU高，基本就是通过线程栈去分析那些线程有问题然后抛给开发去看代码是不是有问题，一个个的去解决，最后成功降低CPU的也不多，堆栈、磁盘I/O、网络这些基本都没怎么重点用到。现在总结下自己面对的问题：①性能分析没有一个整体的分析思路，在遇到各种各样的问题的时候应该用什么样的方法去分析；②自己没有编程开发的经验，代码方面一直是自己的薄弱环节，也在逐渐学习加强。想问下作者，代码开发基础在性能优化工作中到底能占多少比重有多重要？

2018-11-21

作者回复

在分析的过程中一般不需要去做大量开发，随后的优化步骤很可能需要代码开发解决。

2018-11-21



涤生

👍 2

遇到问题时会上网搜索解决办法，但是不知道原理，经常胡乱修改，试错成本太大，希望通过学习提高自己，更好地解决问题

2018-11-21